

Introduksjon til HyperText Markup Language



Skrevet av Atle Nes
Avdeling for informatikk og e-læring
Høgskolen i Sør-Trøndelag

Dette er en kort introduksjon til hvordan du kan lage websider med en enkel teksteditor, slik som NotePad++ - <http://notepad-plus-plus.org/>.

Profesjonelle webutviklere foretrekker vanligvis mer avanserte HTML-editorer, som f.eks. **Adobe Dreamweaver** - <http://www.adobe.com/products/dreamweaver/> eller **Microsoft Expression Web** - http://www.microsoft.com/expression/products/Web_Overview.aspx. Disse kan generere HTML-kode automatisk for deg og man har mulighet til å veksle mellom WYSIWYG-editering, teksteditering og forhåndsvisning. Som nybegynner kan det være en god ide å begynne med en enkel teksteditor for å lære seg språket HTML fra bunnen av.

En god måte å lære HTML på er å se på hvordan andre har kodet sine nettsider. For å gjøre dette må du velge å vise kildekoden til siden i nettleseren din. Du vil nok raskt oppdage at mange nettsider som ser vakre ut kan se ganske rotete ut i koden. Dette kan gi deg et kritisk blikk for hvordan du bør kode din egen side. For å finne fram i uoversiktlig kode finnes det også innebygde verktøy eller tillegg til nettleserne som forenkler denne jobben. For Firefox finnes det f.eks. et tillegg som heter FireBug som lar deg peke på et element på siden og få opp koden som hører til. I Internet Explorer kan du trykke F12 for å få opp utviklerverktøylinjen.

Hva er HTML?

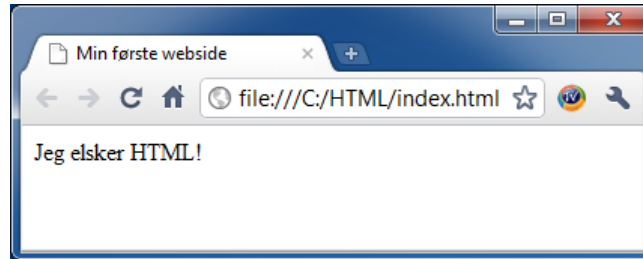
Hyper Text Markup Language (HTML) og er det primære språket som brukes for å lage websider. Det er et markeringsspråk, noe som betyr at man bruker et sett av merker (tags) for å strukturere innholdet i dokumentet. Merkene består av et elementnavn omgitt av vinkelparenteser (<html>) og de kommer vanligvis i par. En kodesnutt åpnes med **startmerket** (<html>) og lukkes med **sluttmerket** (</html>). Mellom paret kan det stå nye merker eller vanlig tekst. Det finnes også **tomme merker**, som f.eks. <hr/> (horisontal delelinje) og
 (linjeskift).

Den viktigste oppgaven til nettleseren er å tolke HTML-dokumenter og vise de fram som en websider. Nettleseren viser da ikke merkene, men bruker merkene for å tolke innholdet i HTML-dokumentet.

Grunnleggende struktur

Når du lagrer et HTML-dokument må det lagres med filendelsen **.htm** eller **.html** for at det skal kjennes igjen av nettleseren. Tidligere var filendelsen begrenset til 3 tegn. I dag er dette ikke noe problem og det anbefales derfor å bruke 4 tegn (.html) som filendelse på HTML-dokumenter.

Alle HTML-dokumenter har en grunnleggende struktur. Vi skal starte med å se på oppbygningen til en svært enkel webside (index.html) med tittelen “Min første webside” som skriver ut teksten “Jeg elsker HTML” i nettleseren. En side som har filnavn **index.html** vil webtjeneren normalt tolke som hovedsiden i en katalog. Dette gjør at man kan slippe unna med å skrive bare katalognavnet i adressen, for eksempel <http://www.hist.no/fag/> i stedet for <http://www.hist.no/fag/index.html>.



Webside: Min første webside (index.html)

Vi kan se på HTML-koden til denne siden ved å velge vis kildekode i nettleseren.

```
<!DOCTYPE html>
<html lang="no">
  <head>
    <title>Min første webside</title>
  </head>

  <body>
    Jeg elsker HTML!
  </body>
</html>
```

Kode: Min første webside (index.html)

Den grunnleggende strukturen kan kort oppsummeres slik:

- Teksten mellom **<html>** og **</html>** definerer hele websiden. Teksten er alltid oppdelt i to deler; hode (**<head>**) og kropp (**<body>**).
- Teksten mellom **<body>** og **</body>** beskriver den synlige delen av websiden.
- Teksten mellom **<head>** og **</head>** inneholder annen informasjon knyttet til websiden, f.eks. tittel på siden.

En tittel

Alle HTML-dokumenter trenger en passende tittel. Tittelen er påkrevd i alle HTML-dokumenter og skal plasseres i head-delen av HTML-dokumentet. Legg merke til at tittelen begynner med startmerket **<title>**, så selve tittelteksten og avsluttes med sluttmerket **</title>**.

Tittelen er et svært viktig element for websidens besøkende, men blir ofte oversett eller glemt av webdesigneren. Dette kan skyldes at tittelen ikke synes på selve websiden.

To grunner til å velge en god tittel:

- Søk er den viktigste måten å finne en nettside på, og en av de viktigste måtene å navigere rundt på nettstedet. Tittelen på siden er ditt beste verktøy for å tiltrekke deg nye besøkende, og å hjelpe eksisterende brukere å finne fram. Tittelen med lenke til nettsiden er så godt som alltid det som vises i søkeresultatet.
- Tittelen brukes på mange steder i nettleserens grensesnitt; som navn øverst på selve nettleservinduet, som navn på hver enkelt arkfane (tab) og i nettleserens historikk. Tittel og URL blir også lagret når nettsiden blir lagt til som bokmerke eller favoritt.

Hva er en god tittel?

En god tittel betyr at man bør velge tittel som er kort og selvforklarende. **Jan Johansens verden** er et bedre valg av tittel enn **Min hjemmeside**. Tittelen bør som tommelfingerregel ikke være lenger enn 64 tegn. For lange titler blir nemlig automatisk forkortet både av nettleser og søkemotor.

Et selskap sin hovedside bør ha en tittel som inneholder selskapets navn. Man kan eventuelt også inkludere en kort beskrivelse av hva selskapet driver med. For alle undersider på nettstedet bør tittelen inneholde ord som godt beskriver hva man kan forvente å finne av innhold på nettopp denne siden.

Hva bør man unngå?

- **Standardtittel:** Hvis du søker på **Untitled Document** eller **Default Page** vil du bli overrasket over hvor mange nettsider som faktisk heter dette. Mange redigeringsverktøy for HTML-dokumenter har en slik standardtittel. Det er en god regel å endre tittelen vekk fra standardtittel før du publiserer nettsiden.
- **Samme tittel på mange dokumenter** vil føre til problemer for navigasjon og søk. Hver gang brukeren lager et bokmerke vil samme tittel brukes som navn. Når man foretar et søk vil man kunne få opp flere treff med samme tittel.
- **Spesialtegn i tittel** kan føre til problemer. Siden Microsoft som lagrer favoritter som filnavn bør du unngå bruk av spesialtegn slik som kolon, semikolon, skråstrek og vinkelparenteser i tittelen.

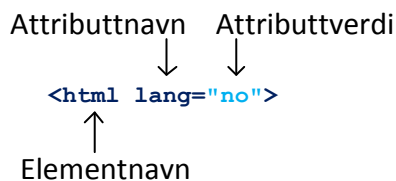
DOCTYPE

Du la kanskje også merke til **DOCTYPE** elementet før `<html>` merket. Dette elementet, som ofte er med først i HTML-koden, gir instruks til nettleseren om hvilket markeringsspråk og eventuelt hvilken versjon av markeringsspråket som brukes i dokumentet. Setningen refererer til en **DTD (Document Type Definition)** hos W3C som inneholder regler for hvordan dokumentet kan bygges opp og hvilke elementer som er tillatt. Dette gjør at nettleserne kan tolke dokumentet og vise fram innholdet på riktig måte. Som du ser av eksemplene er syntaksen svært enkel om du bruker HTML5. Vi har valgt å bruke HTML versjon 5 selv om denne standarden ikke er helt ferdig utviklet. De fleste moderne nettlesere støtter i dag HTML5, og det er dessuten også mulig å legge inn JavaScript som sørger for

en viss bakoverkompatibilitet for eldre nettlesere. En fullstendig oversikt over alle definisjoner i HTML5 finner du blant W3C sine standarder på nett - <http://www.w3.org/TR/html5/>.

Attributter og attributtverdier

Du la kanskje også merke til at det sto litt mer tekst sammen med `<html>` merket i eksempelet vårt. Dette er det som kalles **attributter og attributtverdier**. For HTML-elementet er attributtet lang det eneste som har noen betydning i HTML5. Dette attributtet er valgfritt å ha med og kan brukes til å definere språket som brukes på websiden. Attributtverdien **no** forteller at innholdet er norsk, mens attributtverdien **en** forteller at det er engelsk.



Denne skrivemåten (attributtnavn="attributtverdi") finner vi igjen hos de fleste av elementene i HTML. Alle merker har et antall definerte attributter og attributtverdier. Noen må være med, andre er frivillig å ta med. Man kan ha flere attributter knyttet til samme merke om ønskelig. For en god oversikt over attributter se <http://www.w3schools.com/tags/default.asp>.

Hva er forskjellen på XHTML og HTML?

Du har kanskje hørt om XHTML. XHTML ligner veldig på HTML. Vi kan si at XHTML er mer strukturert enn HTML, ved at språket er redefinert som XML. XHTML-koden må skrives som velformet XML, noe som gjør betyr at syntaksen er strengere og penere enn klassisk HTML. Vi har valgt å bruke XHTML i våre eksempler, men du står fritt til å velge det som passer deg. I HTML forstår nettleseren ofte koden selv om vi glemmer noen merker.

De viktigste forskjellene er:

- XHTML-elementer må nøstes riktig, dvs. merkene må åpnes og lukkes i logisk rekkefølge.
- XHTML-elementer må alltid lukkes, også tomme elementer. Tomme elementer må lukkes med en skråstrek etter elementnavnet, som f.eks. `<hr/>` og `
`.
- XHTML-elementer og attributter må skrives med små bokstaver.
- XHTML-dokumenter skal kun ha et rotelement, som rett og slett er HTML-elementet definert ved merkene `<html>` og `</html>`.
- XHTML-dokumenter kan ha en XML-deklarasjon helt først i fila som inneholder XML-versjonsnummer og hvilket tegnsett som er brukt i dokumentet.

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

- Attributtverdier må være i anførselstegn.
- Attributtverdier kan ikke minimeres. `<option selected>` er et eksempel på et minimert attributt i HTML. I XHTML er tilsvarende uttrykk `<option selected="selected" />`.

Metadata

Innenfor HEAD-elementet kan vi spesifisere en rekke metadata. Dette er data knyttet til websiden som ikke nødvendigvis er av interesse for leseren av siden. Dette kan være f.eks. tegnsett, beskrivelse, nøkkelord, forfatter etc. Siden alle metadata er plassert innenfor HEAD-elementet vil de ikke vises i nettleseren.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <meta name="description" content="Eksempel på bruk av metadata" />
    <meta name="keywords" content="HTML, META, metadata" />
    <meta name="author" content="Atle Nes" />
    <title>Bruk av metadata</title>
  </head>

  <body>
    Jeg elsker metadata!
  </body>
</html>
```

Kode: Bruk av metadata (metadata.html)

Metadata brukes av nettlesere, søkemotorer og webtjenester til å tolke og vise frem websiden. META-elementet kan f.eks. gi deg som webdesigner en mulighet til å påvirke hvordan en webside blir behandlet av søkemotorer og andre tjenester. Det gir deg også muligheten til å la en side og dens lenker bli utelatt fra indeksering av søkemotoren.

```
<meta name="robots" content="noindex, nofollow" />
```

En mer vanlig måte å gjøre dette på er imidlertid å lage seg en robots.txt fil.

Når det gjelder søkemotoroptimalisering har metadata mindre betydning i dag enn tidligere fordi moderne søkemotorer er blitt mer sofistikerte. Mot slutten av 1990-tallet innså selskapene at falske metadata ble lagt inn for at websiden skulle rangeres høyt. Dagens søkemotorer ignorerer derfor nøkkelord, men støtter fortsatt en beskrivelse på inntil 155 tegn.

Metadata for Facebook

Noen webtjenester krever at du legger inn bestemte metadata for at siden skal bli tolket riktig. Et eksempel på en slik webtjeneste er Facebook. Du kan f.eks. bruke følgende kodelinjer for å styre hvilken tittel ("Min tittel"), beskrivelse ("En kort beskrivelse") og miniatyrbilde ("mittbilde.jpg") som vises på Facebook når en bruker deler websiden.

```
<meta property="og:title" content="Min tittel" />
<meta property="og:description" content="En kort beskrivelse" />
<meta property="og:image" content="mittbilde.jpg" />
```

Hvilke metadata som må legges inn i dokumentet må man lese seg til hos hver enkelt tjeneste på nettet.

Tegnsett

For at en webside skal vises riktig må nettleseren vite hva slags tegnsett som er blitt brukt. For sikkerhets skyld bør man derfor alltid ha med hvilket tegnsett man bruker. Et tegnsett forteller hvordan datamaskinen skal tolke en bytekode til riktig tegn. Hvert tegn i tegnsettet har sin unike bytekode og hele dokumentet lagret som en lang rekke med bytekoder.

Hva slags tegnsett som er brukt avgjøres normalt med følgende prioritet:

- 1) Sjekke HTTP Content-Type header fra tjeneren
- 2) Sjekke XML-deklarasjon (XHTML)
- 3) Sjekke HTML/XHTML meta tag
- 4) Bruke andre metoder for å gjette tegnsett

Tidligere var ASCII alfabetet standard tegnsett for HTML. Siden mange land også bruker tegn som ikke er en del av ASCII alfabetet er ISO-8859-1 valgt som standard tegnsett for moderne nettlesere. I oversikten nedenfor kan du for eksempel se at den heksadesimale bytekoden for tegnet Æ er C6.

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SQS	SGCI	SCJ	CSI	ST	QSC	PM	APC
Ax	NBSP	ı	ç	£	¤	¥	ı	§	¨	©	ª	«	¬	SHY	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Oversikt over ISO-8859-1 tegnsettet

Det er ISO (International Standards Organization) som definerer tegnsett for ulike alfabeter og språk. En oversikt over andre tegnsett finner du på http://www.w3schools.com/TAGS/ref_charactersets.asp. Hvis en nettside bruker et annet tegnsett enn ISO-8859-1 bør dette spesifiseres.

Nettopp fordi tegnsettene er begrenset i størrelse og ikke kompatible har Unicode Consortium utviklet det Unicode tegnsettet. Dette er et tegnsett som også er mye i bruk på web. Det er bakoverkompatibelt med ASCII tegnsettet og de første 256 tegnene er lik tegnsettet ISO-8859-1. Dersom du ønsker å bruke UTF-8 på nettsiden i stedet for ISO-8859-1 må du bytte til følgende linje.

```
<meta charset="utf-8" />
```

Kommentarer

Kommentarer kan være nyttige for vedlikehold av nettsiden. De vil ikke bli behandlet av nettleseren, og kan derfor brukes til å gi beskrivende tekst til HTML-koden eller til å kommentere ut HTML-kode.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Min første webside</title>
  </head>

  <body>
    <!-- Dette er en kommentar -->

    <!--
      Kommentarer kan også
      gå over flere linjer
    -->
  </body>
</html>
```

Kode: Eksempel på kommentarer (kommentarer.html)

Avsnitt

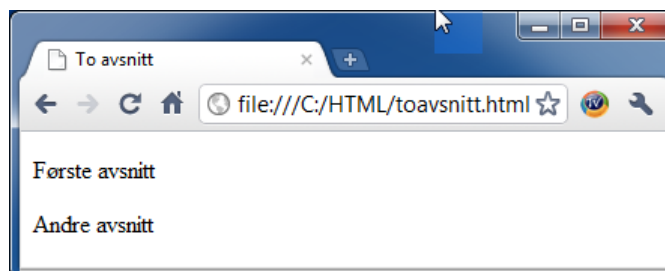
Hvert nytt avsnitt du skriver skal starte med **<p>** merket og avsluttes med **</p>** merket.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>To avsnitt</title>
  </head>

  <body>
    <!-- To avsnitt -->
    <p>Første avsnitt</p>
    <p>Andre avsnitt</p>
  </body>
</html>
```

Kode: Eksempel på avsnitt (toavsnitt.html)



Webside: Eksempel på avsnitt (toavsnitt.html)

Overskrifter

Hvis du har brukt en moderne tekstbehandler slik som Microsoft Word er du kanskje kjent med de innebygde stilene for overskrifter. I HTML finnes seks nivåer av overskrifter fra **<h1>** som er størst, **<h2>** er noe mindre, osv. helt ned til **<h6>** som er minst. Dette kan brukes til å strukturere siden. Søkemotorer bruker også overskrifter for å indeksere innholdet.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

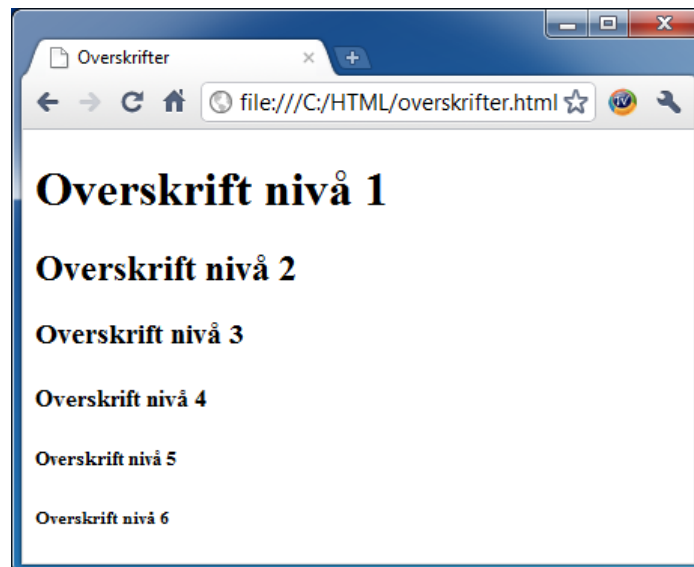
<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Overskrifter</title>
  </head>

  <body>

    <!-- Overskriftene h1 til h6 -->
    <h1>Overskrift nivå 1</h1>
    <h2>Overskrift nivå 2</h2>
    <h3>Overskrift nivå 3</h3>
    <h4>Overskrift nivå 4</h4>
    <h5>Overskrift nivå 5</h5>
    <h6>Overskrift nivå 6</h6>

  </body>
</html>
```

Kode: Eksempel på overskrifter (overskrifter.html)



Webside: Eksempel på overskrifter (overskrifter.html)

Uthevet, kursiv og overstreket tekst

Noen ganger er det ønskelig ha annet utseende på teksten. Uthevet og kursiv skrift brukes for å fremheve deler av teksten. Du kan få til uthvet skrift ved å bruke **** merket og kursiv skrift ved å bruke **<i>** merket. Overstreket tekst brukes ofte dersom man gjør rettelser på en side, men ønsker å vise at her er det gjort rettelser. Da kan man bruke **<s>** merket.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

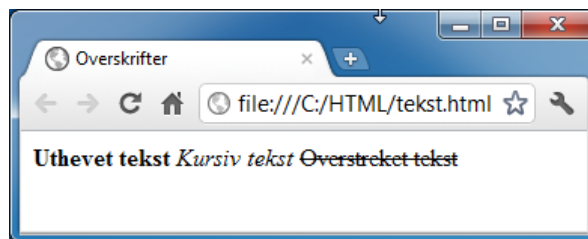
<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Overskrifter</title>
  </head>

  <body>

    <!-- Overskriftene h1 til h6 -->
    <b>Uthevet tekst</b>
    <i>Kursiv tekst</i>
    <s>Overstreket tekst</s>

  </body>
</html>
```

Kode: Eksempel på tekstmanipulasjon (tekst.html)



Webside: Eksempel på tekstmanipulasjon (tekst.html)

Lenker

Hyperlenker er en av de viktigste funksjonene ved websider. De muliggjør referanser til andre ressurser. Dette kan være andre dokumenter, bilder etc. For å lenke til en annen webside bruker vi koden `navn` der lenke er URL til siden og navn er teksten som synes på websiden. Lenka kan være relativ eller absolutt avhengig av om det er en ressurs på samme webtjener eller ikke. Om man ønsker å la lenka sprette opp i et nytt vindu eller en ny fane kan du legge til attributtet `target="_new"`.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Fire lenker</title>
  </head>

  <body>
    <h1>Her er mine nettaviser</h1>
    <p>Klikk på navnet for å gå til siden</p>

    <!-- Fire lenker -->
    <p><a href="http://www.vg.no"/>Verdens Gang</a></p>
    <p><a href="http://www.db.no"/>Dagbladet</a></p>
    <p><a href="http://www.ap.no"/>Aftenposten</a></p>
    <p><a href="http://www.adressa.no"/>Adresseavisen</a></p>

  </body>
</html>
```

Kode: Eksempel på lenker (firelenker.html)



Webside: Eksempel på lenker (firelenker.html)

E-post

Det går også an å lenke til e-postadresser ved å skrive f.eks. **mailto:ola.nordmann@norge.no** som attributtverdi i stedet for en hyperlenke. Når noen klikker på en slik lenke vil nettleseren normalt åpne standard e-postprogram og opprette en ny utgående e-postmelding. Denne metoden anbefales ikke om du vil la besøkende sende e-post fra siden. Spammere har nemlig roboter som søker nettet automatisk etter e-postadresser ved hjelp av mailto-attributtverdien og krøllalfategnet (@). Et alternativ kan være å legge inn epostadressen på siden som et bilde.

Destinasjonsanker

Det går også an å lage lenker ikke bare til andre websider, men også lenker internt på siden. Dette gjør man ved først å definere seg ulike destinasjonsankere rundt om på siden. Vi kan som eksempel lage ankeret kapittel2 på siden ved å skrive **** på et vilkårlig sted på siden og en lenke dit fra en helt annen plass med ****. Dette kan være spesielt nyttig om du har en veldig lang webside. Nettleseren vil da scrolle dokumentet automatisk til det forhåndsdefinerte destinasjonsankeret når du klikker på lenka.

Om du vil kan du selvfølgelig kombinere både lenking til en annen webside og et destinasjonsanker på denne websiden. Da skriver du f.eks. **** for å hoppe til ankeret regler på side2.html.

Bilder

Hittil har vi bare tatt for oss tekst. De fleste websider inneholder både tekst og bilder.

De tre mest populære bildeformatene som brukes av webutviklere er:

- 1) Graphics Interchange Format (GIF)
- 2) Joint Photographic Experts Group (JPEG)
- 3) Portable Network Graphics (PNG)

Disse formatene er støttet av alle moderne nettlesere. For å sette inn et bilde på en webside bruker vi koden **** der bildefil er URL til bildet. Det er anbefalt at bilder ligger lagret i samme katalog eller på samme webtjener, men de kan være lagret på en helt annen webtjener på nettet. Ulempen med lenking til en ekstern ressurs er at ressursen kan forsvinne. Legg også merke til at dette er et IMG er et tomt element og normalt avsluttes uten sluttmerke.

Det er vanlig å ha med noen flere attributter når vi legger inn bilder på webtjeneren. De to vanligste er høyde (**height**) og bredde (**width**) definert som piksler. Disse kan brukes til å få nettleseren til å vise fram bildet i en annen størrelse enn den faktiske størrelsen til bildet.

XHTML-standarder krever dessuten at man har med attributtet **alt** som inneholder en alternativ tekst som kan vises fram dersom bildet av en eller annen grunn ikke kan vises fram. Det kan skyldes at brukeren har en nettleser som ikke støtter bilder, f.eks. en tekstbasert nettleser som Lynx. Attributtet er også viktig for mennesker med synshemming som får teksten lest opp for seg. Da vil programmet kunne lese opp teksten i stedet for å vise fram bildet.

Bilder som lenker

Det er selvfølgelig også å bruke bilder som hyperlenker. Det gjør vi ved å sette et IMG element innenfor et A element. Koden `` setter inn bildefil på websiden og sier at dersom du klikker på bilde vil du bli sendt til URL i lenke. Dette kan brukes til å lage grafiske menyer.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>To bilder</title>
  </head>

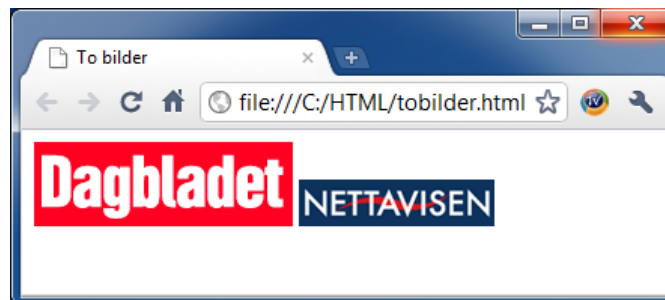
  <body>

    <!-- To bilder med lenke -->
    <a href="http://www.db.no/">
      
    </a>

    <a href="http://www.nettavisen.no/">
      
    </a>

  </body>
</html>
```

Kode: Eksempel på bildelenker (tobilder.html)



Webside: Eksempel på bildelenker (tobilder.html)

Legg merke til at vi i tillegg har lagt til attributtet `border="0"` for å fjerne blå ramme rundt bildet. Dette er en ramme som dukker opp for å vise at bildet lenker til en webside. Denne effekten er vanligvis ikke noe vi ønsker å se.

Trykkfølsomme bilder

Det er også mulig å lage seg bilder der det kan klikkes i soner på selve bilder og hver av sonene er lenker til individuelle lenker.

Vi kan f.eks. tenke oss et bilde som forestiller en knapperad:

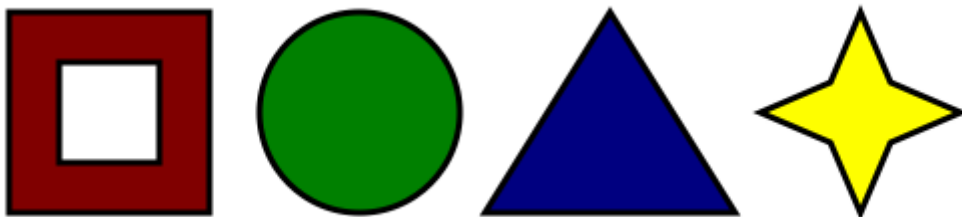


Først må vi definere et kart over knappene (sonene i bildet):

```
<map name="knappene">
  <area shape="rect" coords="0,0,50,40" href="back.html" />
  <area shape="rect" coords="50,0,100,40" href="forward.html" />
  <area shape="rect" coords="100,0,150,40" href="home.html" />
  <area shape="rect" coords="0,0,500,40" href="all.html" />
</map>
```

For å ta det i bruk skriver vi ``. Kartet kan plasseres på vilkårlig sted i koden innenfor body, men en god regel er å plassere det sammen med bildet. Vi kan se at hvert av områdene har fått definert lenker til nye sider (back.html, forward.html og home.html). I tillegg er det laget et område for hele bildet som blir brukt dersom vi klikker utenfor de tre mindre områdene som er definert. For å finne koordinatene er det mange programmer som kan gjøre dette for deg. Noen genererer hele koden for deg. Du kan f.eks. bruke MS Paint og notere koordinatene når du fører musen over bildet.

Det er mulig å lage mer avanserte kart ved å bruke andre former. Vi kan f.eks. tenke oss et bilde med en rød boks, en grønn sirkel, en blå trekant og en gul stjerne:



Koden kan i dette tilfellet se slik ut:

```

<map name="shapes">
  <area shape="rect" coords="50,50,100,100"/> <!-- hullet i rød boks -->
  <area shape="rect" coords="25,25,125,125" href="red.html" alt="rød boks"/>
  <area shape="circle" coords="200,75,50" href="green.html" alt="grønn
sirkel"/>
  <area shape="poly" coords="325,25,262,125,388,125" href="blue.html"
alt="blå trekant"/>
  <area shape="poly"
coords="450,25,435,60,400,75,435,90,450,125,465,90,500,75,465,60"
href="yellow.html" alt="gul stjerne"/>
</map>
```

Lister

En måte å organisere informasjon på en webside på er å bruke lister. De to viktigste typene vi finner i XHTML er ordnet og uordnet liste.

Uordnet liste

Uordnet liste er en liste der innholdet ikke er ordnet etter nummer eller bokstav. En slik liste starter med `` og avsluttes med ``. Hvert element i listen starter med `` og avsluttes med ``. De fleste moderne nettlesere viser dette fram som en punktliste med hvert element på en ny linje og et kulepunkt framfor hvert element.

Ordnet liste

Ordnet liste er en liste der innholdet er ordnet etter nummer eller bokstav. En slik liste starter med `` og avsluttes med ``. Hvert element i listen starter med `` og avsluttes med `` som for uordnet liste. De fleste moderne nettlesere viser dette fram som en liste nummerert fra 1 og oppover med hvert element på en ny linje og nummeret framfor hvert element.

Det er mulig å legge til attributter på ol merket for å starte på et annet nummer, f.eks. attributtet **start="5"** for å starte nummereringen på fem, eller velge annen nummertype, f.eks. små romertall med attributtet **type="i"**.

Nøstede lister

Lister kan nøstes inni hverandre. Et eksempel på dette er vist nedenfor.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

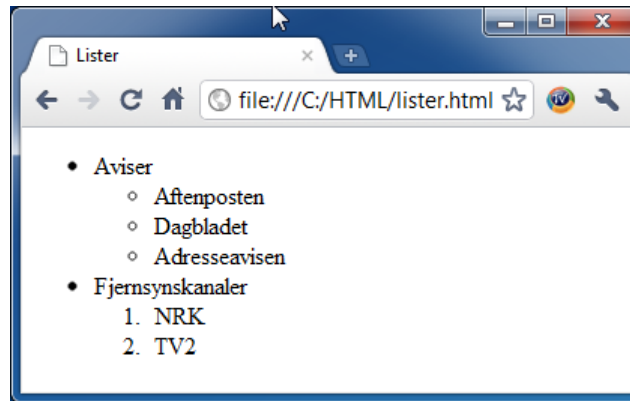
<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Lister</title>
  </head>
  <body>

    <ul> <!-- Uordnet liste -->
      <li>Aviser
        <ul> <!-- Nøstet uordnet liste -->
          <li>Aftenposten</li>
          <li>Dagbladet</li>
          <li>Adresseavisen</li>
        </ul>
      </li>

      <li>Fjernsynskanaler
        <ol> <!-- Nøstet ordnet liste -->
          <li>NRK</li>
          <li>TV2</li>
        </ol>
      </li>
    </ul>

  </body>
</html>
```

Kode: Eksempel på lister (lister.html)



Webside: Eksempel på lister (lister.html)

Tabeller

En annen måte å organisere informasjon på en webside på er å bruke tabeller. Tabeller består av celler definert ved rader og kolonner. Tabeller har tradisjonelt vært brukt til å presentere data på tabellform, men også til å styre layout på websiden. Det vil si at man har delt opp siden i rader og kolonner og plassert innhold på siden ved å legge innholdet i riktig celle på siden. Det anbefales i dag at layout hovedsakelig spesifiseres med CSS i stedet for med tabeller.

Dette avsnittet om tabeller vil derfor bare omhandle bruk av tabeller til presentasjon av data. Alle tabeller starter med **<table>** og avsluttes med **</table>**. Attributtet **border** definerer tykkelsen på tabellens ramme i piksler. For å lage en tabell uten rammer setter du **border="0"**. Et annet mye brukt attributt er **width** som setter tabellens bredde. For å lage en tabell som er 300 piksler bred kan du sette **width="300"**. Det går også an å sette bredden til 40% av nettleserens bredde ved å skrive **width="40%"**. Prøv å endre størrelsen på nettleserens vindu og se hvordan dette påvirker tabellen. For større avstand mellom hver celle kan du bruke attributtet **cellspacing**, mens attributtet **cellpadding** kan brukes til å øke avstanden mellom celleveggen og cellens innhold.

På samme måte som alt attributtet for bilder finnes det måter å beskrive innholdet i tabellen. Dette kan for eksempel gjøres med CAPTION elementet.

En tabell har tre deler; head, body og foot. Head starter med **<thead>** og avsluttes med **</thead>**, body starter med **<tbody>** og avsluttes med **</tbody>** og foot starter med **<tfoot>** og avsluttes med **</tfoot>**. Head inneholder vanligvis informasjon slik som kolonnenavn, body inneholder tabellens primære data, mens foot vanligvis inneholder resultater og fotnoter. Alle disse tre delene kan inneholde rader og kolonner. Det er imidlertid bare body som er nødvendig for å lage en tabell. Legg også spesielt merke til at foot må defineres før body i koden, selv om den vises fram nederst i tabellen.

En rad starter med **<tr>** og slutter med **</tr>**. Hver rad har et antall kolonner som starter med **<td>** og slutter med **</td>**. Informasjonen innenfor hvert kolonneelement er således definert ved en rad og kolonne og er det som vises fram i hver enkeltcelle i tabellen. I head og foot starter kolonnene med **<th>** og slutter med **</th>**. De fleste moderne nettlesere viser tekst i head og foot sentrert med uthevet skrift.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html>

<html lang="no">
  <head>
    <meta charset="iso-8859-1" />
    <title>Tabell</title>
  </head>

  <body>

    <!-- Tabell med head, body og foot -->
    <table border="1" width="40%">
      <caption>Prisen på frukt</caption>

      <thead> <!-- Kolonneoverskrifter først i tabellen -->
        <tr>
          <th>Frukt</th>
          <th>Pris</th>
        </tr>
      </thead>

      <tfoot> <!-- Oppsummering sist i tabellen -->
        <tr>
          <th>Totalt</th>
          <th>15 kr</th>
        </tr>
      </tfoot>

      <tbody> <!-- Innholdet midt i tabellen -->
        <tr>
          <td>Eple</td>
          <td>5 kr</td>
        </tr>
        <tr>
          <td>Appelsin</td>
          <td>10 kr</td>
        </tr>
      </tbody>

    </table>
  </body>
</html>

```

Kode: Eksempel på tabell (tabell.html)



Frukt	Pris
Eple	5 kr
Appelsin	10 kr
Totalt	15 kr

Webside: Eksempel på tabell (tabell.html)

Dersom man ønsker å bygge mer innviklede tabeller kan man bruke **rowspan** og **colspan** attributtene for å slå sammen celler over flere rader eller flere kolonner. Dersom man skriver f.eks. **<td rowspan="2">** betyr dette at cellen spenner over to rader i tabellen, og skriver man **<td colspan="4">** betyr det at cellen spenner over fire kolonner i tabellen.

Spesialtegn

Noen tegn kan ikke brukes direkte, men må skrives på en spesiell måte for å bli tolket rett. Dette gjelder f.eks. mellomrom som skrives som kan settes inn med koden ** ** (non-breaking space), eller æøå som kan settes inn med kodene **æ**, **ø** og **å**. Om du har problemer med spesialtegn eller vil være helt sikker på at spesialtegnene vises rett bør du bruke denne skrivemåten i koden din.

Mer om spesialtegn kan du lese på:

- http://www.w3schools.com/tags/ref_ascii.asp
- http://www.w3schools.com/tags/ref_symbols.asp